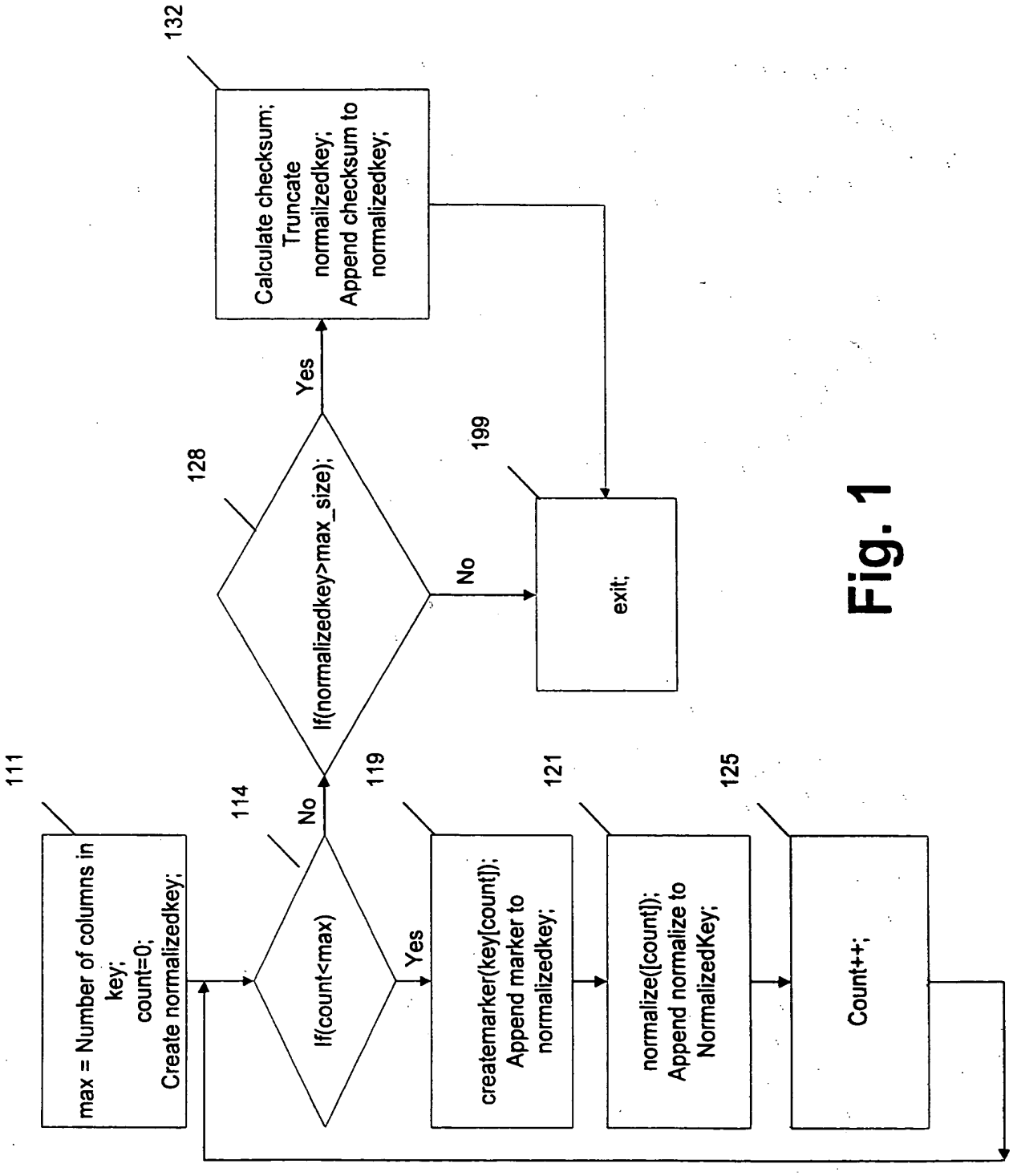
111
max = Number of columns in key;
count=0;
Create normalizedkey;
114
If(count<max)
No
Yes
119
createmarker(key[count]);
Append marker to normalizedkey;
121
normalize([count]);
Append normalize to NormalizedKey;
125
Count++;
128
If(normalizedkey>max_size);
Yes
No
132
Calculate checksum;
Truncate normalizedkey;
Append checksum to normalizedkey;
199
exit;

Fig. 1

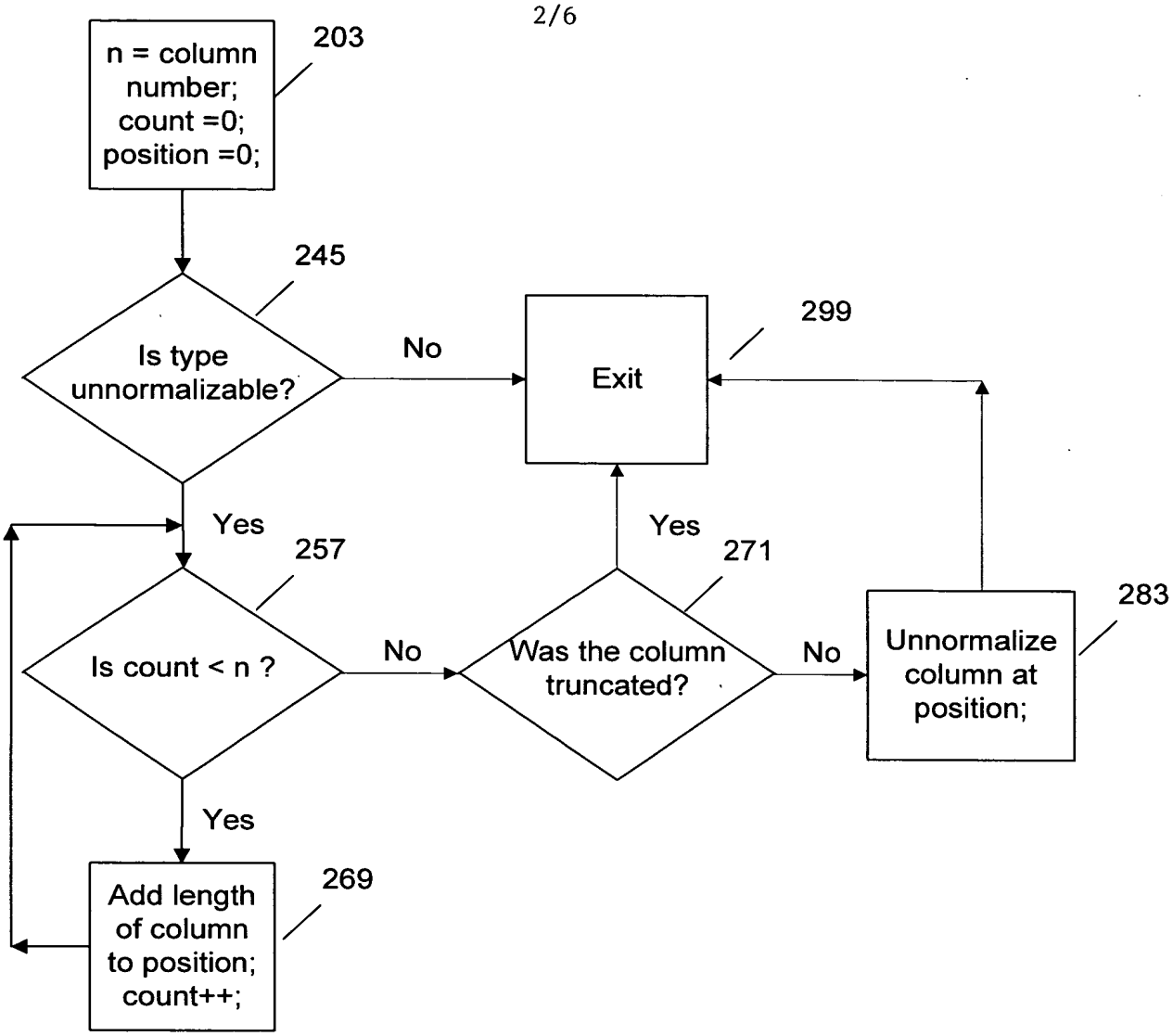
n = column number; count =0; position =0;
203
245
Is type unnormalizable?
No
Exit
299
Yes
257
Is count < n ?
No
Yes
271
Was the column truncated?
No
Unnormalize column at position;
283
Yes
Add length of column to position; count++;
269

Fig. 2

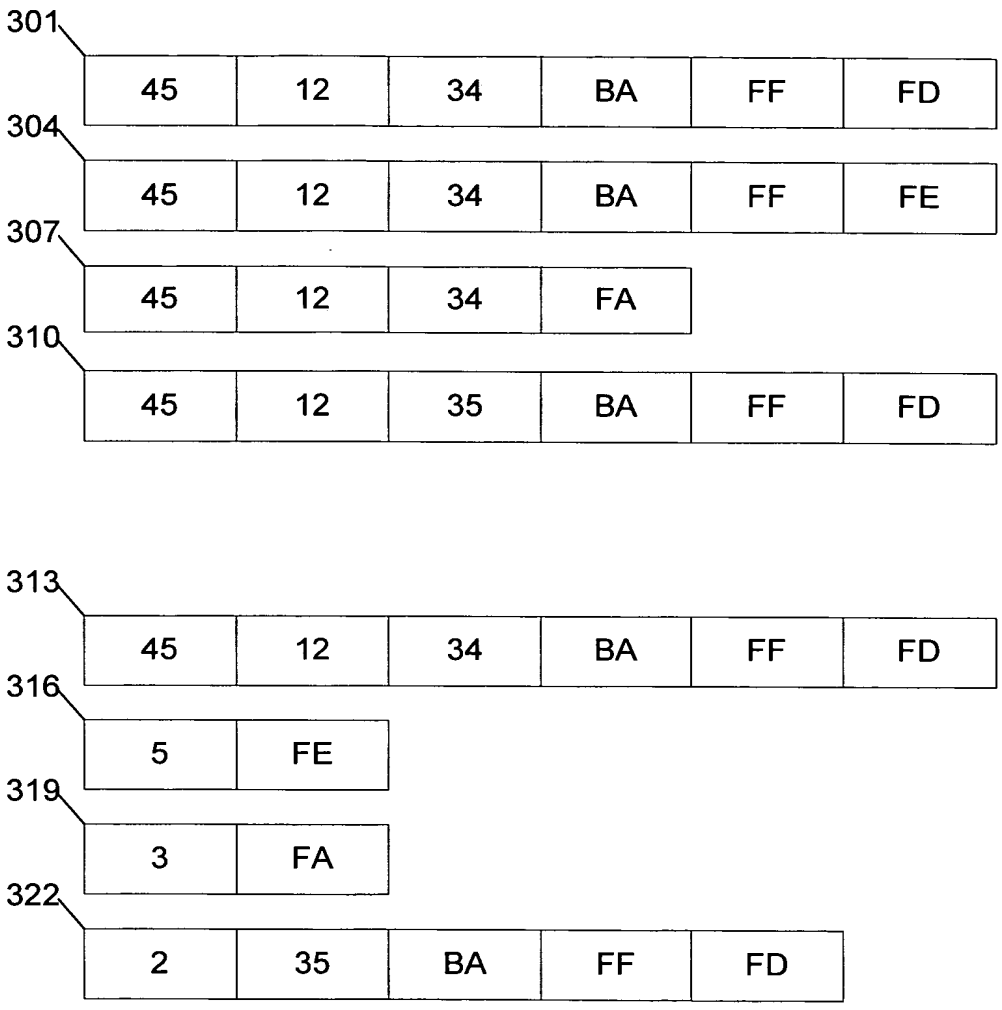
301
45
12
34
BA
FF
FD
304
45
12
34
BA
FF
FE
307
45
12
34
FA
310
45
12
35
BA
FF
FD
313
45
12
34
BA
FF
FD
316
5
FE
319
3
FA
322
2
35
BA
FF
FD

Fig. 3

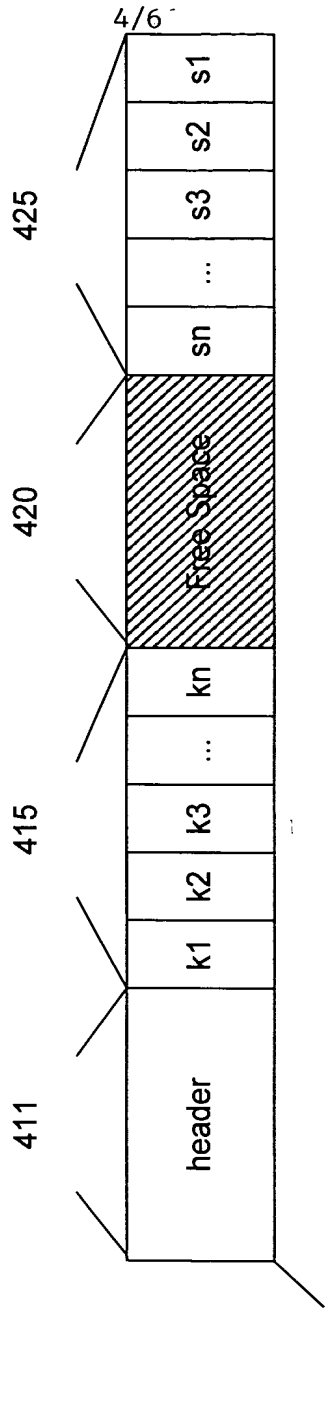

Fig. 4

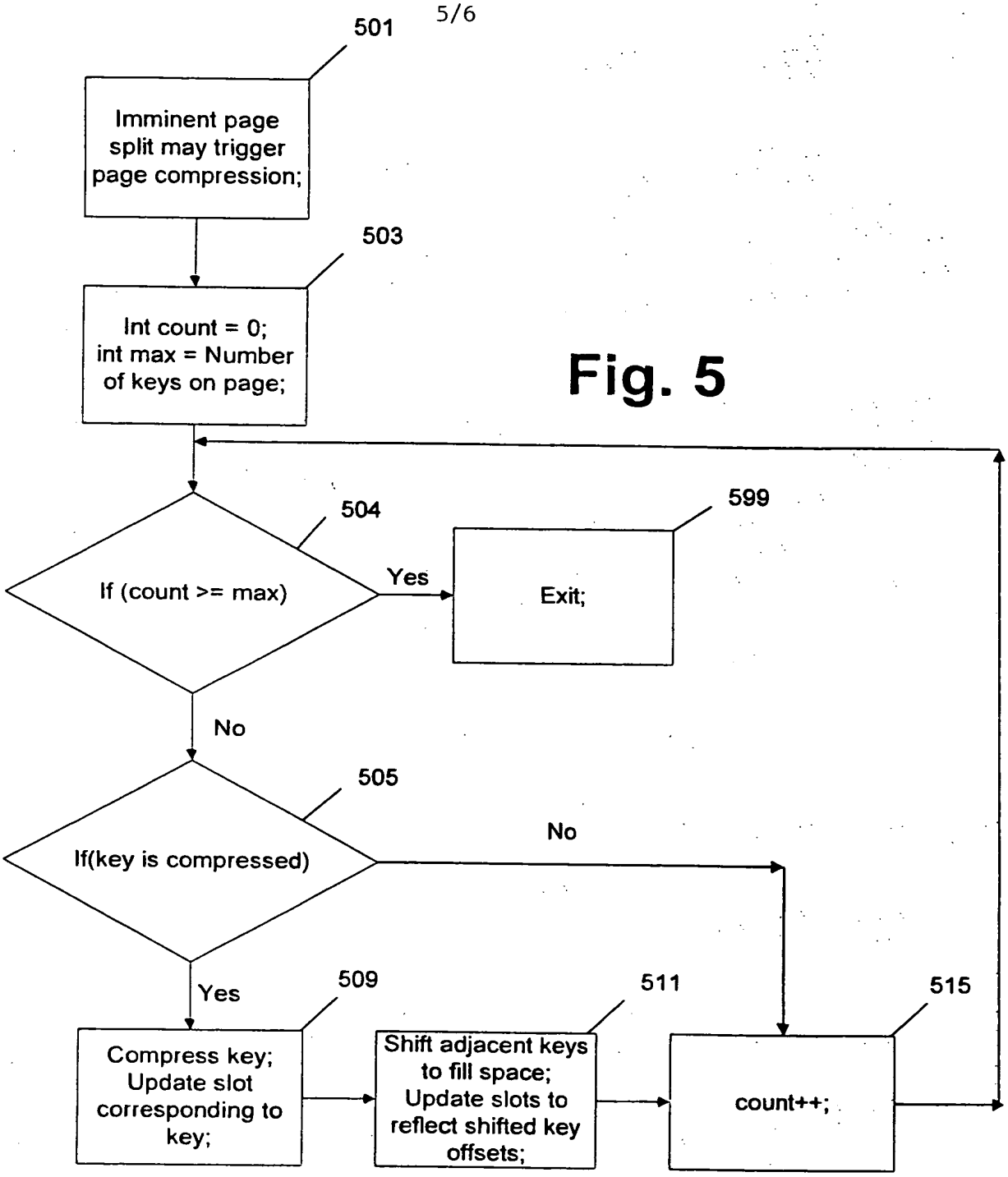
501
Imminent page split may trigger page compression;
503
Int count = 0; int max = Number of keys on page;
504
If (count >= max)
Yes
599
Exit;
No
505
If(key is compressed)
No
Yes
509
Compress key; Update slot corresponding to key;
511
Shift adjacent keys to fill space; Update slots to reflect shifted key offsets;
515
count++;

Fig. 5

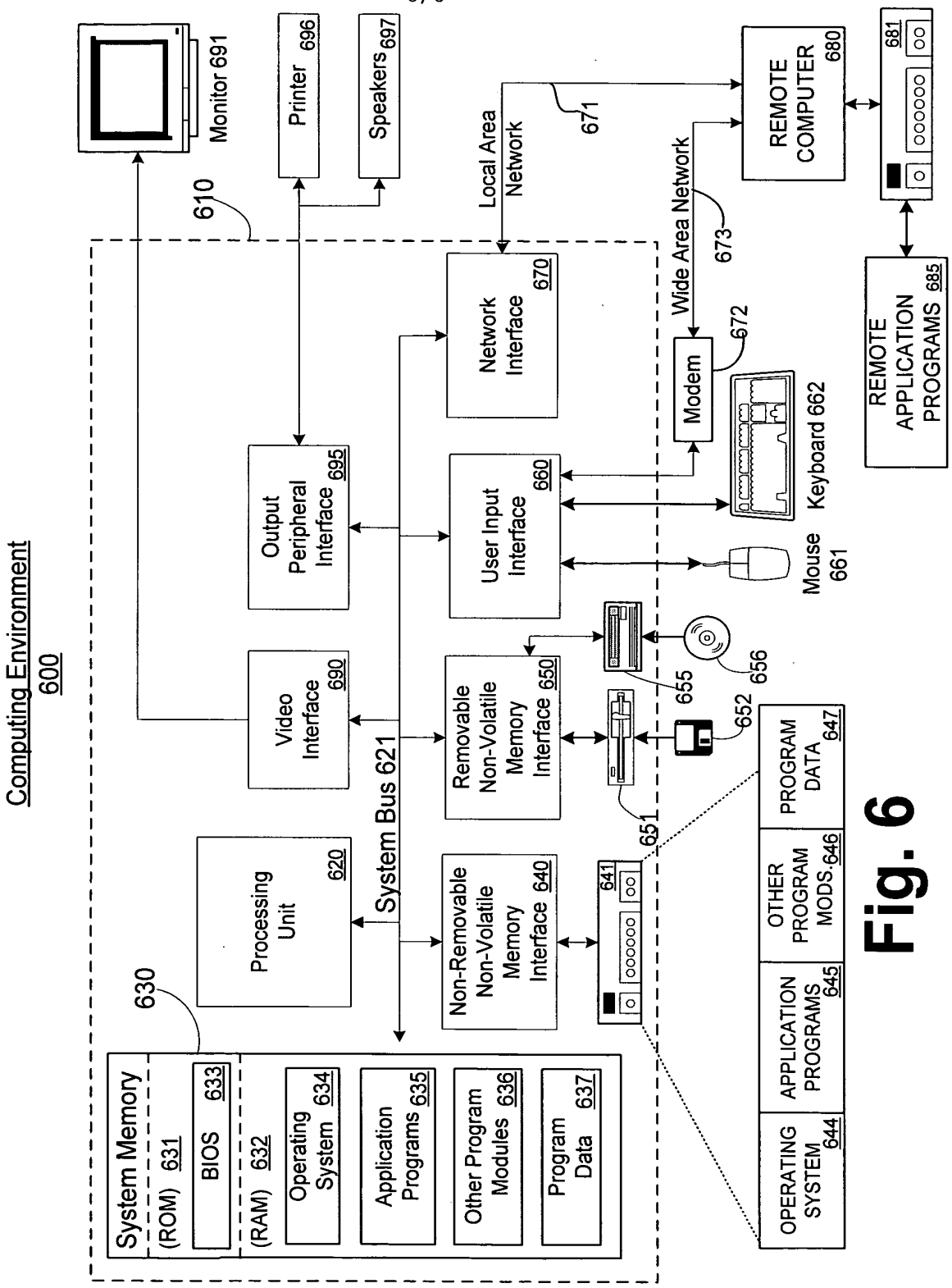
Computing Environment
600
610
Monitor 691
Printer 696
Speakers 697
Local Area Network
671
REMOTE COMPUTER 680
681
Wide Area Network
673
Network Interface 670
Modem 672
REMOTE APPLICATION PROGRAMS 685
Output Peripheral Interface 695
User Input Interface 660
Keyboard 662
Mouse 661
Video Interface 690
Removable Non-Volatile Memory Interface 650
655
656
652
651
System Bus 621
Processing Unit 620
Non-Removable Non-Volatile Memory Interface 640
641
630
System Memory
(ROM) 631
BIOS 633
(RAM) 632
Operating System 634
Application Programs 635
Other Program Modules 636
Program Data 637
OPERATING SYSTEM 644
APPLICATION PROGRAMS 645
OTHER PROGRAM MODS 646
PROGRAM DATA 647

Fig. 6